

# Linear-Time Algorithms for the Farthest-Segment Voronoi Diagram and Related Tree Structures<sup>\*</sup>

Elena Khramtcova and Evanthia Papadopoulou

Faculty of Informatics, Università della Svizzera italiana (USI), Lugano, Switzerland  
{elena.khramtcova, evanthia.papadopoulou}@usi.ch

**Abstract.** We present linear-time algorithms to construct tree-like Voronoi diagrams with disconnected regions, after the sequence of their faces along an enclosing boundary (or at infinity) is known. We focus on Voronoi diagrams of line segments including the farthest-segment Voronoi diagram, the order- $(k+1)$  subdivision within an order- $k$  Voronoi region, and deleting a site from a nearest-neighbor segment Voronoi diagram. Although tree-structured, these diagrams illustrate properties surprisingly different from their counterparts for points. The sequence of their faces along the relevant boundary forms a Davenport-Schinzel sequence of order  $\geq 2$ . Once this sequence is known, we show how to compute the corresponding Voronoi diagram in linear time, expected or deterministic, augmenting the existing frameworks for points in convex position with the ability to handle non-point sites and multiple Voronoi faces. Our techniques contribute towards the question of linear-time construction algorithms for Voronoi diagrams whose graph structure is a tree.

## 1 Introduction

It is well known that the Voronoi diagram of points in convex position can be computed in linear time, given the order of their convex hull [1]. The same holds for a class of related diagrams such as the farthest-point Voronoi diagram, computing the medial axis of a convex polygon, and deleting a point from the nearest-neighbor Voronoi diagram. In an abstract setting, a *Hamiltonian abstract Voronoi diagram* [10] can be computed in linear time, given the order of Voronoi regions along an unbounded simple curve, which visits each region exactly once and can intersect each bisector only once. This has been recently extended to include forest structures [5] under the same conditions of no repetition of regions and bisectors along the curve. The medial axis of a simple polygon can also be computed in linear time [8]. It is therefore natural to ask what other types of Voronoi diagrams can be constructed in linear time. In this paper we address generalized tree-like Voronoi diagrams that have disconnected regions.

---

<sup>\*</sup> Research supported in part by the Swiss National Science Foundation, project 20GG21-134355, under the ESF EUROCORES program EuroGIGA/VORONOI.

Classical variants of Voronoi diagrams such as higher-order Voronoi diagrams for sites other than points, had surprisingly been ignored in the literature of computational geometry until recently [4,14]. Given a set  $S$  of  $n$  simple geometric objects in the plane, called sites, the *order- $k$  Voronoi diagram* of  $S$  is a partitioning of the plane into regions such that every point within a region has the same  $k$  nearest sites. For  $k = 1$ , this is the *nearest-neighbor Voronoi diagram* and for  $k = n - 1$  it is the *farthest-site Voronoi diagram* of  $S$ . Despite similarities, these diagrams for non-point sites, e.g., line segments, illustrate fundamental structural differences from their counterparts for points, such as the presence of disconnected regions (see also [2,6,11]). This had been a gap in the computational geometry literature, until recently, as segment Voronoi diagrams are fundamental to problems involving proximity of polygonal objects. This paper contributes further in closing this gap. For more information on Voronoi diagrams see the book of Aurenhammer et al. [3].

In this paper we give linear-time algorithms (expected and deterministic) for constructing tree-like Voronoi diagrams with disconnected regions, after the sequence of their faces within an enclosing boundary or at infinity is known. We focus on the farthest-segment Voronoi diagram, however, the same techniques are applicable to constructing the order- $(k+1)$  subdivision within a given order- $k$  segment Voronoi region and updating in linear time the nearest-neighbor segment Voronoi diagram following deletion of one site. Interestingly, the latter two problems require computing initially two different tree-like diagrams. A major difference from the respective problems for points is that the sequence of faces along the relevant enclosing boundary forms a Davenport-Schinzel sequence of order at least two,<sup>1</sup> in contrast to the case of points, where no repetition can exist. Repetition introduces several complications, including the fact that the sequence of Voronoi faces along the relevant boundary for a subset of the original segments,  $S' \subset S$ , is not a subsequence of the respective sequence for  $S$ . In addition, such a subsequence may not even correspond to a Voronoi diagram. Thus, intermediate diagrams of our algorithms are interesting on their own right. They have the structural properties of the relevant segment Voronoi diagram, however, they do not correspond to such a diagram nor are they instances of abstract Voronoi diagrams.

The purpose of this paper is to extend the paradigm of the existing linear constructions for tree-structured diagrams beyond the case of points in convex position [1]. Our goal is to generalize fundamental techniques known for points to more general objects so that the computation of their basic diagrams can be unified, despite their structural differences. As a byproduct we also improve the time complexity of the basic iterative approach to construct the order- $k$  Voronoi diagram to  $O(k^2n + n \log n)$  from the standard  $O(k^2n \log n)$  [14], and also updating a nearest neighbor diagram after deletion of one site in time proportional to the updates performed in the diagram.

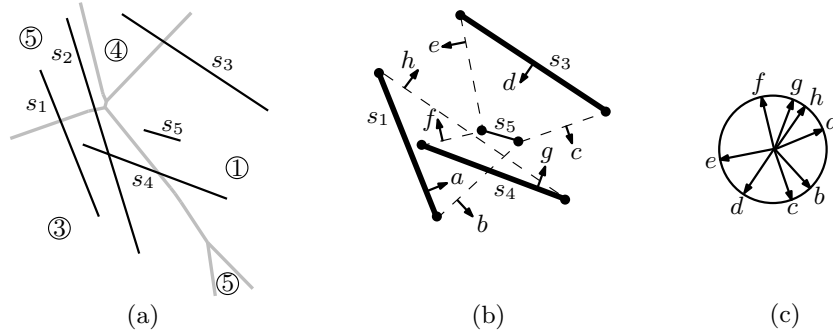
<sup>1</sup> Order-3 for the farthest-segment Voronoi diagram, order-4 for the order- $k$  segment Voronoi diagram, order-2 for disjoint segments or the corresponding abstract Voronoi diagrams.

The segment counterpart of order- $k$  Voronoi diagrams,  $k \geq 1$ , is an important variant for a variety of applications dealing with polygonal objects or embedded planar graphs, see, e.g., [12] and references therein, for examples in the semiconductor industry.

## 2 Preliminaries and Definitions

Let  $S$  be a set of arbitrary line segments in  $\mathbb{R}^2$ ; segments in  $S$  may intersect or touch at a single point. The distance between a point  $q$  and a line segment  $s_i$  is  $d(q, s_i) = \min\{d(q, y), y \in s_i\}$ , where  $d(q, y)$  denotes the ordinary distance between two points  $q, y$  in the  $L_2$  (or the  $L_p$ ) metric. The bisector of two segments  $s_i, s_j \in S$  is  $b(s_i, s_j) = \{x \in \mathbb{R}^2 \mid d(x, s_i) = d(x, s_j)\}$ . For disjoint segments,  $b(s_i, s_j)$  is an unbounded curve that consists of a constant number of pieces, where each piece is a portion of an elementary bisector between the endpoints and open portions of  $s_i, s_j$ . If two segments intersect at point  $p$ , their bisector consists of two such curves intersecting at  $p$ .

The farthest Voronoi region of a segment  $s_i$  is  $\text{freq}(s_i) = \{x \in \mathbb{R}^2 \mid d(x, s_i) > d(x, s_j), 1 \leq j \leq n, j \neq i\}$ . For disjoint line segments or line segments that intersect but do not touch at endpoints, the order- $k$  Voronoi region of a set  $H$ , where  $H \subset S$ ,  $|H| = k$ , and  $1 \leq k \leq n - 1$ , is  $k\text{-reg}(H) = \{x \mid \forall s \in H, \forall t \in S \setminus H, d(x, s) < d(x, t)\}$ . For an extension of this definition to line segments forming a *planar straight-line graph*, see [14]. Note, for  $k = n - 1$ ,  $\text{freq}(s_i) = k\text{-reg}(S \setminus \{s_i\})$ . The non-empty farthest (resp., order- $k$ ) Voronoi regions of all the segments in  $S$ , together with their bounding edges and vertices, define a partition of the plane, called the *farthest-segment Voronoi diagram*  $\text{FVD}(S)$ , see Fig. 1(a), (resp., order- $k$  Voronoi diagram). Any maximally-connected subset of a Voronoi region is called a *face*.



**Fig. 1.** [13] (a)  $\text{FVD}(S)$ ,  $S = \{s_1, \dots, s_5\}$ ; (b) its farthest hull; (c)  $\text{Gmap}(S)$

A farthest Voronoi region  $\text{freq}(s_i)$  is non-empty and unbounded in direction  $\phi$  if and only if there exists an open halfplane, normal to  $\phi$ , which intersects all segments in  $S$  but  $s_i$  [2]. The line  $\ell$ , normal to  $\phi$ , bounding such a halfplane, is

called a *supporting line*. The direction  $\phi$  (normal to  $\ell$ ) is referred to as the *hull direction* of  $\ell$  and it is denoted by  $\nu(\ell)$ . An unbounded Voronoi edge separating  $\text{freg}(s_i)$  and  $\text{freg}(s_j)$  is a portion of  $b(p, q)$ , where  $p, q$  are endpoints of  $s_i$  and  $s_j$ , such that the line through  $\overline{pq}$  induces an open halfplane that intersects all segments in  $S$ , except  $s_i, s_j$  (and possibly except additional segments incident to  $p, q$ ). Segment  $\overline{pq}$  is called a *supporting segment*, and the direction normal to it pointing to the inside of this halfplane is denoted by  $\nu(\overline{pq})$  and is called the *hull direction* of  $\overline{pq}$ . A segment  $s_i \in S$  such that the line  $\ell$  through  $s_i$  is supporting, is called a *hull segment*; its *hull direction* is  $\nu(s_i) = \nu(\ell)$ , which is normal to  $\ell$ . Figs. 1(a) and 1(b) illustrate a farthest-segment Voronoi diagram and its hull respectively. In Fig. 1(b), supporting segments are shown in dashed lines, and hull segments are shown in bold. Arrows indicate the hull directions of all supporting and hull segments of  $S$ . The closed polygonal line obtained by following the supporting and hull segments in the angular order of their hull directions (shown in e.g., Fig. 1(c)) is called the *farthest hull*. For more information see [13].

The Gaussian map of FVD( $S$ ), denoted as Gmap( $S$ ) (see Fig. 1(c)) provides a correspondence between the faces of FVD( $S$ ) and a circle of directions  $K$  [13]. Each Voronoi face is mapped to an arc on  $K$ , which represents the set of directions along which the face is unbounded. The Gmap( $S$ ) can be viewed as a cyclic sequence of arcs, where each arc corresponds to one face of FVD( $S$ ). Two neighboring arcs  $\alpha, \gamma$  are separated by the hull direction  $\nu(\alpha, \gamma)$ , which corresponds to a supporting segment  $\overline{pq}$  ( $\nu(\alpha, \gamma) = \nu(\overline{pq})$ ) as well as to an unbounded Voronoi edge, which is a portion of  $b(\alpha, \gamma)$  (i.e., a portion of  $b(p, q)$ ). There are two types of arcs: (1) arcs that correspond to a single endpoint of a segment, called a *single-vertex* arc; and (2) arcs that correspond to an entire segment (a hull segment), called a *segment arc*. A segment arc consists of two sub-arcs, one for each endpoint of the segment, separated by the segment's hull direction. The Gmap( $S$ ) can be computed in  $O(n \log n)$  time (or output-sensitive  $O(n \log h)$ , where  $h = |\text{Gmap}(S)|$ ) [13].

The standard point-line duality transformation  $T$  offers a correspondence between the faces of FVD( $S$ ) and envelopes of *wedges* [2]. A segment  $s_i = uv$  is sent into a *lower wedge*, defined by the lower envelope of  $T(u)$  and  $T(v)$  (see, e.g., Fig. 6), and to an *upper wedge* defined as the area above the upper envelope of  $T(u)$ ,  $T(v)$ . Let  $E$  (resp.,  $E'$ ) be the boundary of the union of the lower (resp., upper) wedges. The faces of FVD( $S$ ) in cyclic order, correspond exactly to the edges of  $E$  and  $E'$  in  $x$ -order [2]. Thus, there is a clear correspondence between  $E$  (resp.,  $E'$ ) and the *upper* (resp., *lower*) Gmap: the vertices of  $E$  are exactly the hull directions of supporting segments on the upper Gmap and the apexes of wedges in  $E$  are exactly the hull directions of hull segments [13]. The edges of  $E$  in increasing  $x$ -order correspond exactly to the arcs of the *upper Gmap* in counterclockwise order. The upper and lower Gmap is the portion of Gmap( $S$ ) above and below the horizontal diameter of  $K$  respectively. In fact, any  $x$ -monotone path  $p$  in the arrangement of upper (resp., lower) wedges can be transformed into a sequence of arcs in the *upper* (resp., *lower*) circle of directions,

which is the portion of  $K$  above (resp., below) its horizontal diameter. Each edge of  $p$ , portion of  $T(u)$ , corresponds to an arc of  $u$  on the upper (resp., lower)  $K$ , and each vertex of  $p$ , which is an intersection point  $T(u) \cap T(v)$ , corresponds to  $\nu(\overline{uv})$ , where  $\nu(\overline{uv})$  is the hull direction of supporting segment  $\overline{uv}$  that separates the respective arcs of  $u$  and  $v$  on the upper (resp., lower)  $K$ .

Throughout this paper, given an arc  $\alpha$ , let  $s_\alpha$  denote the segment in  $S$  that induces  $\alpha$ .

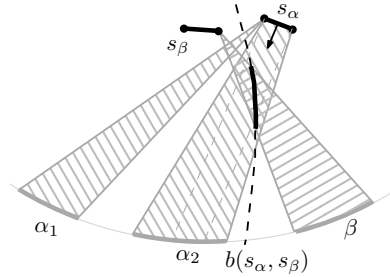
### 3 The Farthest Voronoi Diagram of a Sequence

Let  $G$  be a cyclic sequence of arcs on the circle of directions  $K$ , which corresponds to a pair of  $x$ -monotone paths in the dual space, one in the arrangement of upper wedges and one in the arrangement of lower wedges. For brevity,  $G$  is simply called an arc sequence. Any consecutive arcs in  $G$  of the same segment are assumed to be united into a single maximal arc.

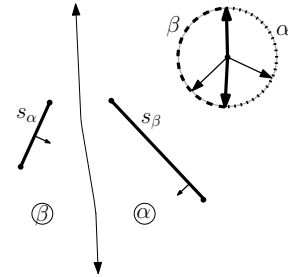
We define the farthest Voronoi diagram of such an arc sequence  $G$ , which is  $\text{FVD}(S)$  for  $G = \text{Gmap}(S)$ . The diagrams of such sequences appear as intermediate diagrams in the process of computing  $\text{FVD}(S)$ , given  $\text{Gmap}(S)$ , but they do not correspond to any type of segment Voronoi diagram. We first define such a diagram and then illustrate issues related to the deletion and insertion of arcs.

Given an arc  $\alpha \in G$  and a point  $x \in \mathbb{R}^2$ ,  $x \notin s_\alpha$ , let  $r(x, s_\alpha)$  denote the unbounded portion of the ray that realizes the Euclidean distance from  $s_\alpha$  to  $x$ , starting at  $x$  and extending to infinity away from  $s_\alpha$ . We say that  $x$  is *attainable* from  $\alpha$  if the direction of  $r(x, s_\alpha)$  is contained in  $\alpha$ . A point  $x \in \alpha$  inducing  $\alpha$  is always *attainable* from  $\alpha$ . Let  $d(x, \alpha) = d(x, s_\alpha)$ , if  $x$  is attainable from  $\alpha$ , and let  $d(x, \alpha) = -\infty$ , otherwise. The locus of points attainable from arc  $\alpha$  is called the *attainable region* of  $\alpha$ ,  $R(\alpha)$ . Fig. 2 illustrates shaded the attainable regions of arcs  $\alpha_1, \alpha_2$ , and  $\beta$ . Intuitively, arc  $\alpha$  *exists* only within its attainable region.

*Remark 1.* For arcs  $\alpha_1, \alpha_2 \in G$  of the same segment  $s_\alpha$ ,  $R(\alpha_1) \cap R(\alpha_2) \setminus \{s_\alpha\} = \emptyset$ .



**Fig. 2.** Attainable regions  $R(\alpha_1), R(\alpha_2), R(\beta)$



**Fig. 3.**  $\text{FVD}(G)$ ,  $|G| = 2$

Given two arcs  $\alpha, \beta$  ( $s_\alpha \neq s_\beta$ ) we define their *arc bisector* as  $b(\alpha, \beta) = b(s_\alpha, s_\beta) \cap R(\alpha) \cap R(\beta)$ , which is a single connected component of  $b(s_\alpha, s_\beta)$ . If

$s_\alpha = s_\beta$ , then  $b(\alpha, \beta)$  is not defined, unless  $\alpha$  and  $\beta$  are consecutive, in which case  $b(\alpha, \beta) = R(\alpha) \cap R(\beta)$  is the ray bounding their neighboring attainable regions, called an *artificial bisector* (see Def. 2).

The farthest Voronoi region of an arc  $\alpha$  is now defined in the ordinary way

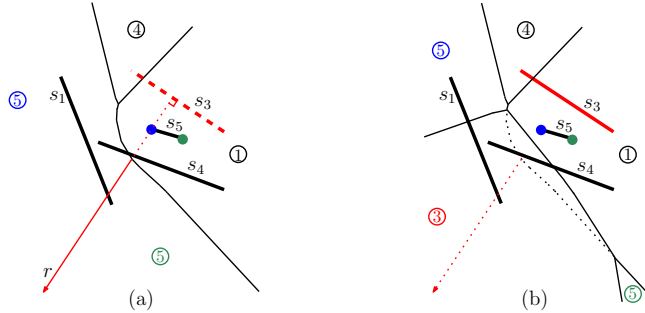
$$\text{freg}(\alpha) = \{x \in \mathbb{R}^2 \mid d(x, \alpha) > d(x, \gamma), \forall \text{ arc } \gamma \in G, \gamma \neq \alpha\}.$$

The subdivision of the plane derived by these regions and their boundaries, for all arcs in  $G$ , is called the *farthest Voronoi diagram* of  $G$ , denoted  $\text{FVD}(G)$ . The closure of  $\text{freg}(\alpha)$  is denoted by  $\overline{\text{freg}(\alpha)}$ .

**Definition 1.** Let  $\mathcal{T}(G) = \mathbb{R}^2 \setminus \cup_{\alpha \in G} \text{freg}(\alpha)$ . If all edges of  $\mathcal{T}(G)$  are portions of arc bisectors, then  $G$ ,  $\mathcal{T}(G)$ , and  $\text{FVD}(G)$  are all called proper.

When  $\mathcal{T}(G)$  is proper, it is simply the graph structure of  $\text{FVD}(G)$ . For an arbitrary arc sequence, however,  $\mathcal{T}(G)$  may contain boundaries of attainable regions and even two-dimensional regions (i.e.,  $\text{FVD}(G)$  may contain holes). The diagrams and sequences produced by our algorithms are always proper.

Fig. 4(a) illustrates  $\text{FVD}(G)$  for a proper arc sequence  $G$  derived from the  $\text{Gmap}(S)$  of Fig 1, consisting of three maximal arcs of segments  $s_1, s_4, s_5$ . The maximal arc of segment  $s_5$  consists of two different arcs of  $\text{Gmap}(S)$ . Ray  $r$  indicates an *artificial bisector* of these two arcs that separates their corresponding regions (see Def. 2). Fig. 4(b) illustrates  $\text{FVD}(G'')$ , where  $G''$  contains an additional arc  $\beta$  of segment  $s_3$  ( $G'' = G \oplus \beta$ ).



**Fig. 4.**  $\text{FVD}(G)$  for an arc sequence of Fig. 1; (a)  $\text{FVD}(G)$ ; (b)  $\text{FVD}(G'')$ ,  $G'' = G \oplus \beta$ .

**Lemma 1.** For a proper arc sequence  $G$ ,  $\mathcal{T}(G)$  is a (connected) tree.

*Proof.* Suppose that  $G$  is proper, that is, all the edges of  $\mathcal{T}(G)$  are portions of arc bisectors. Let  $x$  be a point on  $\mathcal{T}(G)$  along arc bisector  $b(\alpha, \beta)$ . We first prove that the entire unbounded ray  $r(x, s_\alpha)$  must be enclosed in  $\overline{\text{freg}(\alpha)}$ .

Since  $b(\alpha, \beta)$  is an arc bisector, a neighborhood of  $r(x, s_\alpha)$  near  $x$  must be contained in  $\text{freg}(\alpha)$ . But no arc bisector involving  $\alpha$  can bound  $r(x, s_\alpha)$  as we walk on it starting at  $x$ , unless an arc  $\delta$  suddenly becomes attainable because  $r(x, s_\alpha)$  intersects  $R(\delta)$  at point  $z$  and  $d(z, \delta) > d(z, \alpha)$ . But then  $z \in \mathcal{T}(G)$

without being on an arc bisector, contradicting our assumption about  $\mathcal{T}(G)$ . Thus, the entire ray  $r(x, s_\alpha) \in \overline{freg(\alpha)}$ .

It remains to show that  $\mathcal{T}(G)$  is connected. Suppose otherwise. Then there must be non-consecutive arcs  $\alpha_1, \alpha_2$  such that  $freg(\alpha_1)$  and  $freg(\alpha_2)$  are path-connected, which could only occur for arcs of the same segment  $s_\alpha$ . This means that for any point  $p$  in  $freg(\alpha_1)$  and point  $q$  in  $freg(\alpha_2)$ , there is a path  $\pi$  connecting them such that  $\pi$  is entirely contained in  $freg(\alpha_1) \cup freg(\alpha_2)$ . But then for any point  $x$  along  $\pi$ , ray  $r(x, s_\alpha)$  would be entirely contained in  $freg(\alpha_1) \cup freg(\alpha_2)$ , i.e., arcs  $\alpha_1, \alpha_2$  would be neighboring, a contradiction. (See also Remark 1.) Thus,  $\mathcal{T}(G)$  is connected and no two regions of (non-consecutive) arcs of the same segment can be neighboring.  $\square$

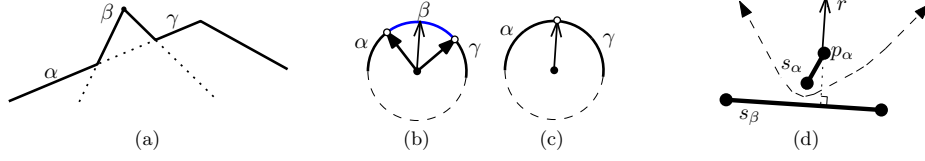
An arc sequence  $G$  is called a *subsequence* of  $\text{Gmap}(S)$  if every arc of  $G$  entirely contains a corresponding arc of  $\text{Gmap}(S)$  induced by the same segment. The arcs in  $G$  are simply expanded versions of the arcs in  $\text{Gmap}(S)$ . The arcs in  $\text{Gmap}(S)$  as well as their expanded versions in  $G$  are called *original arcs*. A sequence  $G'$  is called an *augmented subsequence* of  $\text{Gmap}(S)$  if  $G'$  contains at least one (expanded) arc of  $\text{Gmap}(S)$  for every segment that appears in  $G'$ . Thus, an augmented subsequence contains *original arcs*, which are expanded versions of the arcs in  $\text{Gmap}(S)$ , and *new arcs*, which do not correspond to arcs of  $\text{Gmap}(S)$ . An augmented subsequence  $G'$ , which has the same original arcs as a subsequence  $G$ , is said to be *corresponding* to  $G$ .

Note that in dual space,  $G$  and  $G'$  no longer correspond to envelopes of wedges, but to  $x$ -monotone paths that contain portions of these envelopes. Assuming lower wedges, the path of an augmented subsequence  $G'$  is above (or on) the path of the corresponding subsequence  $G$ . Note also that a subsequence of  $\text{Gmap}(S)$  need not be proper. Our algorithms deal with augmented subsequences so that they become proper.

### 3.1 Deletion and insertion of arcs

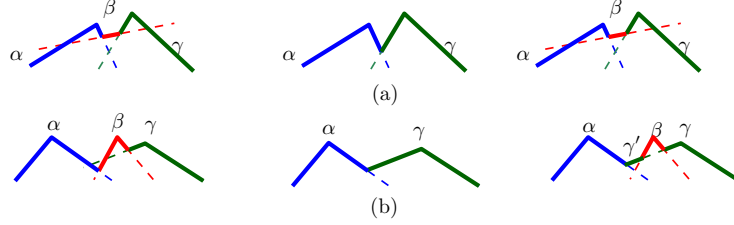
Throughout our algorithms we use a deletion and insertion process of arcs in sequences derived from  $\text{Gmap}(S)$ . The deletion process produces subsequences of  $\text{Gmap}(S)$  that are not necessarily proper. Thus, the insertion process introduces new arcs, creating augmented subsequences that remain proper. New arcs clearly cannot appear in the final diagram. Let  $G \ominus \beta$  (resp.,  $G \oplus \beta$ ) denote the arc sequence obtained from  $G$  after deleting from it (resp., inserting to it) arc  $\beta$ . In fact,  $G \oplus \beta$  is the arc sequence obtained by inserting  $freg(\beta)$  to  $\text{FVD}(G)$ .

*Arc deletion.* A subsequence  $G$  is derived from  $\text{Gmap}(S)$  by deletion of arcs. When deleting an arc  $\beta$ , the neighboring arcs  $\alpha$  and  $\gamma$  *expand* over  $\beta$  (see Fig. 5(a)-(c)). Either both  $\alpha$  and  $\gamma$  expand (see e.g., Fig. 5 or Fig. 6(a) for segments in the dual space) or one expands while the other shrinks (see Fig. 6(b)). During the expansion,  $\alpha$  and  $\gamma$  may change from being a single-vertex arc to a segment arc. Since  $\alpha$  and  $\gamma$  are original, they both remain present in  $G \ominus \{\beta\}$ ,



**Fig. 5.** Sequence  $\alpha\beta\gamma$ ,  $s_\alpha = s_\gamma$ . (a) The dual wedges; (b)  $G$ ; (c)  $G \ominus \{\beta\}$ ; (d) The artificial bisector  $b(\alpha, \gamma) = r$ ; the dashed curve indicates  $b(s_\alpha, s_\beta)$ .

and their common endpoint becomes  $\nu(\alpha, \gamma)$ . Assuming  $s_\alpha \neq s_\gamma$ ,  $\nu(\alpha, \gamma)$  corresponds to bisector  $b(\alpha, \gamma)$  as obtained  $b(s_\alpha, s_\gamma)$ . If  $s_\alpha = s_\gamma$ , let  $\nu(\alpha, \gamma) = \nu(s_\beta)$ . If  $s_\alpha = s_\beta$  then  $\alpha$  expands to cover the entire  $\beta$  and  $\nu(\alpha, \gamma) = \nu(\beta, \gamma)$ .



**Fig. 6.** Deleting and re-inserting  $\beta$  in sequence  $\alpha\beta\gamma$ . (a)  $\alpha$  and  $\gamma$  enlarge; (b)  $\gamma$  enlarges,  $\alpha$  shrinks. From left to right: the initial sequence; after deleting  $\beta$ ; after re-inserting  $\beta$ .

Given an (augmented) subsequence  $G$ , consider the result of inserting back  $\beta$  between (the expanded) arcs  $\alpha, \gamma$  (see Fig. 6 for an illustration in dual space). The result is  $\alpha\beta\gamma$ , if both  $\alpha, \gamma$  expanded when removing  $\beta$ , or  $\alpha\gamma'\beta\gamma$ , if  $\alpha$  shrunk, or  $\alpha\beta\alpha'\gamma$ , if  $\gamma$  shrunk, where  $\alpha'$  and  $\gamma'$  are *new* arcs of  $s_\alpha$  and  $s_\gamma$  respectively. Re-inserting  $\beta$  in subsequence  $G \ominus \beta$  results in an augmented subsequence  $G' = (G \ominus \beta) \oplus \beta$ , which may be different from  $G$ .

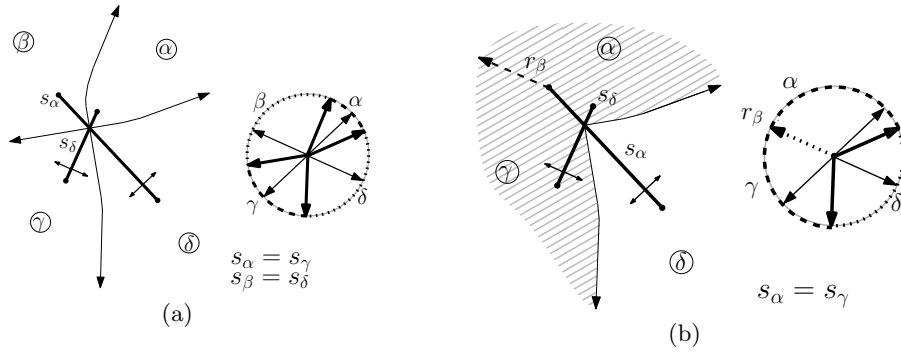
**Definition 2.** Let  $\alpha, \gamma$  be consecutive arcs in  $G$ ,  $s_\alpha = s_\gamma$ . Let  $\beta$  be the arc whose removal made  $\alpha$  and  $\gamma$  consecutive. The artificial bisector  $b(\alpha, \gamma)$  is (or contains) a ray in the direction of  $\nu(s_\beta)$ , emanating from the relevant endpoint  $p_\alpha$  of  $s_\alpha$  (see Fig. 5(d)) and  $\nu(\alpha, \gamma) = \nu(s_\beta)$ . Formally,  $b(\alpha, \gamma) = R(\alpha) \cap R(\gamma)$ , where  $\nu(\alpha, \gamma) = \nu(s_\beta)$ .

*Arc insertion.* Let  $G'$  be a proper augmented subsequence of  $\text{Gmap}(S)$  and let  $\beta$  be an original arc,  $\beta \notin G'$ . Let  $\alpha, \gamma$  be two consecutive original arcs in  $G'$ , such that  $\beta$  is between them in  $\text{Gmap}(S)$ . Because of new arcs, arcs  $\alpha$  and  $\gamma$  need not be neighbors in  $G'$ . To insert  $\beta$  between  $\alpha$  and  $\gamma$  in  $G'$  there are several cases to be considered. They include: (1)  $s_\alpha \neq s_\beta \neq s_\gamma$  and  $\nu(\alpha, \gamma)$  in  $\beta$  (this is the standard case, see Figs. 6(a) and 8); (2)  $s_\alpha = s_\gamma$  ( $\beta$  is inserted over the artificial bisector  $b(\alpha, \gamma)$ , represented by  $\nu(\alpha, \gamma)$ , see Fig. 4); (3) Arc  $\alpha$  (equiv,



$\gamma$ ), as it appears expanded in  $G'$ , entirely contains  $\beta$  (see Fig.6(b); the insertion of  $\beta$  splits  $\alpha$  and a new arc  $\alpha'$  is created; i.e.,  $\text{freq}(\beta)$  splits  $\text{freq}(\alpha)$  in two, where one part is  $\text{freq}(\alpha')$ ); (4)  $s_\alpha = s_\beta$  (equiv.  $s_\beta = s_\gamma$ ) ( $\alpha$  is simply split in two by  $\nu(\alpha, \beta)$ ; i.e.,  $\text{freq}(\alpha)$  is split in two by the artificial bisector  $b(\alpha, \beta)$  and one part becomes  $\text{freq}(\beta)$ ). Some consecutive new arcs between  $\alpha, \gamma$  may be deleted by the insertion of  $\beta$ . More details are given in *phase 2* of the randomized algorithm in Sec. 4.

Assuming that  $G'$  is proper, after the insertion of  $\beta$ ,  $G'' = G' \oplus \beta$  remains a proper augmented subsequence of  $\text{Gmap}(S)$  (see Lemma 2).



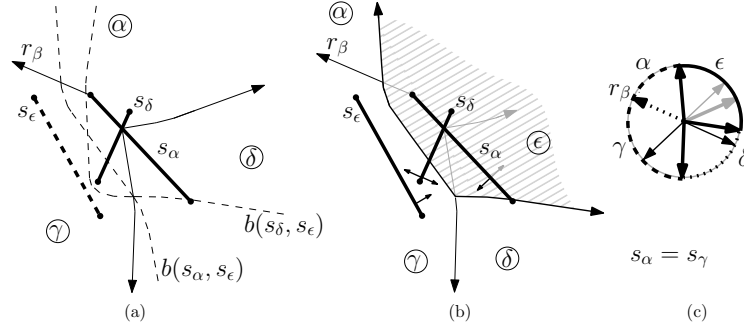
**Fig. 7.**  $\text{FVD}(G)$ ,  $S = \{s_\alpha, s_\delta\}$ ; (a)  $G = \text{Gmap}(S)$ ; (b)  $G = \alpha\gamma\delta = \text{Gmap}(S) \ominus \beta$

Figs. 3 and 7 illustrate arc sequences of two segments and their Voronoi diagrams. In Fig. 3 the segments are disjoint and  $G = \text{Gmap}(S)$ . In Fig 7 the segments intersect; Fig 7(a), illustrates  $\text{FVD}(S)$  and  $\text{Gmap}(S)$ ; Fig 7(b) illustrates  $\text{FVD}(G)$  for a subsequence  $G$  obtained by deleting arc  $\beta$  from  $\text{Gmap}(S)$ . By deleting  $\beta$ , arcs  $\alpha$  and  $\gamma$  of segment  $s_\alpha = s_\gamma$  become neighbors, corresponding to one maximal arc  $\alpha\gamma$  whose region is shown shaded in Fig 7(b). Region  $\text{freq}(\alpha\gamma)$  is split into subregions  $\text{freq}(\alpha)$  and  $\text{freq}(\gamma)$  by the artificial bisector  $b(\alpha, \gamma) = r_\beta \cup s_\alpha$ ;  $r_\beta$  (shown dashed) follows the hull direction of  $s_\beta = s_\delta$ . Note that if  $p_\alpha \in \text{freq}(\alpha\gamma)$ , as in Fig 7(b), then  $b(\alpha, \gamma)$  includes also  $s_\alpha$ .

Fig. 8 illustrates the insertion of an arc  $\epsilon$  of segment  $s_\epsilon$  into  $\text{FVD}(G)$ ,  $G = \alpha\gamma\delta$ , of Fig. 7(b). Fig. 8(a) shows  $\text{FVD}(G)$  superimposing segment  $s_\epsilon$ ,  $b(s_\alpha s_\epsilon)$  and  $b(s_\delta, s_\epsilon)$  in dashed lines. Fig. 8(b) shows  $\text{FVD}(G'')$ ,  $G'' = \alpha\gamma\delta\epsilon$ , where  $\text{freq}(\epsilon)$  is shaded. Gray lines show the deleted parts of  $\text{FVD}(G)$  and  $G$  respectively.

We remark that the artificial bisector  $b(\alpha, \gamma)$  is not uniquely defined but it depends on the arc  $\beta$  whose deletion made  $\alpha, \gamma$  consecutive. It is used to split  $\text{freq}(\alpha\gamma)$  into subregions  $\text{freq}(\alpha)$  and  $\text{freq}(\gamma)$  so that  $\alpha, \gamma$  can be treated as different arcs. No portion of  $b(\alpha, \gamma)$  can remain in the diagram after the insertion of arc  $\beta$  because for any point  $y$  along  $b(\alpha, \gamma)$ ,  $d(y, \beta) = d(y, s_\beta) > d(y, s_\alpha) = d(y, \alpha\gamma)$ .

Note that any arc sequence  $G, |G| \geq 2$ , of two segments is proper (see Figs. 3,7).



**Fig. 8.** Insertion of arc  $\epsilon$  in  $\text{FVD}(G)$ ,  $G = \alpha\gamma\delta$ ; (a)  $\text{FVD}(G)$ ; (b)  $\text{FVD}(G'')$ ,  $G'' = \alpha\gamma\delta\epsilon$ ; (c)  $G''$

**Lemma 2.** *Let  $G'$  be a proper augmented subsequence of  $\text{Gmap}(S)$  and let  $\beta$  be an original arc,  $\beta \notin G'$ . The sequence  $G'' = G' \oplus \beta$  obtained by the insertion of  $\beta$  in  $G'$  is a proper augmented subsequence of  $\text{Gmap}(S)$ .*

*Proof.* Since  $\text{FVD}(G')$  is a proper diagram it is enough to show that  $\partial \text{freq}(\beta)$  in  $\text{FVD}(G'')$  consists solely of arc bisectors, which implies that all edges of  $\mathcal{T}(G'')$  must be arc bisectors.

There are three cases: (1)  $\text{freq}(\beta)$  is inserted between the region of  $\alpha$  and  $\gamma$ , where  $s_\alpha \neq s_\beta \neq s_\gamma$  ( $\alpha, \gamma$  may be original or new arcs). (2)  $\text{freq}(\beta)$  is inserted within  $\text{freq}(\alpha)$  splitting it into two regions. This is when the original arc  $\beta$  (as it appears in  $\text{Gmap}(S)$ ) is enclosed in  $\alpha$  (as it appears in  $G'$ ). (3)  $s_\alpha = s_\beta$  (resp.,  $s_\gamma = s_\beta$ ). By definition,  $\text{freq}(\beta)$  is entirely contained in  $R(\beta)$ .

In the first two cases the unbounded portion of  $\partial \text{freq}(\beta)$  are clearly well defined arc bisectors. Consider the boundary of  $R(\beta)$ , which consists of two rays emanating from  $s_\beta$ , one in the direction of  $\nu(\alpha, \beta)$  and one in the direction of  $\nu(\beta, \gamma)$ . Because of well known visibility properties of segment bisectors,  $b(\alpha, \beta)$  cannot hit the boundary of  $R(\beta)$  neighboring  $\alpha$ , and  $b(\beta, \gamma)$  cannot hit the boundary of  $R(\beta)$  neighboring  $\gamma$ . (Note that ray  $r(x, s_\alpha)$ ,  $x \in b(\alpha, \beta)$ , lies entirely in one side of  $b(\alpha, \beta)$ .) Furthermore, since  $\text{freq}(\alpha)$  and  $\text{freq}(\gamma)$  are regions of  $\text{FVD}(G')$ , which is proper,  $b(\alpha, \beta)$  cannot hit the boundary of  $R(\alpha)$  and  $b(\beta, \gamma)$  cannot hit the boundary of  $R(\gamma)$ . Thus, at least one of  $b(\alpha, \beta)$  or  $b(\beta, \gamma)$  must end at a proper Voronoi vertex. The same argument repeats for the next Voronoi edge along  $\partial \text{freq}(\beta)$  incident to this proper Voronoi vertex. Thus, all vertices along  $\partial \text{freq}(\beta)$  must be proper and all edges must be arc bisectors.

In case (3),  $\alpha\beta$  is one maximal arc, thus,  $\text{FVD}(G'')$  is essentially the same as  $\text{FVD}(G')$ . To keep  $\alpha$  and  $\beta$  separate we simply split  $\text{freq}(\alpha)$  by inserting the artificial bisector  $b(\alpha, \beta)$ , which must intersect  $\partial \text{freq}(\alpha)$  at a proper Voronoi vertex.  $\square$

We now use the concepts presented in this section to obtain linear-time algorithms (expected and deterministic) to compute  $\text{FVD}(G)$ , given  $\text{Gmap}(S)$ .

## 4 A Randomized Linear Construction

We give an expected linear-time algorithm to compute  $\text{FVD}(S)$ , given  $\text{Gmap}(S)$ . It is inspired by the two-phase randomized approach of [7] for points in convex position and uses the concepts of Sec. 3. We give the randomized algorithm first, because its simplicity helps gain intuition before applying the same techniques in the more involved deterministic framework in Sec. 5.

Let  $\alpha_1, \alpha_2, \dots, \alpha_h$  be a random permutation of arcs in  $\text{Gmap}(S)$ , and let  $A_i = \{\alpha_1, \alpha_2, \dots, \alpha_i\}$ ,  $1 \leq i \leq h$ , be the set of the first  $i$  arcs in this order. Let  $t$  be the largest index such that  $\alpha_1, \dots, \alpha_t$  consists of arcs of only two segments that form exactly two maximal arcs. (Note that an arc sequence of two segments may form 2 to 4 maximal arcs if the segments intersect.)

The algorithm proceeds in two phases. Phase 1 computes the subsequence  $G_i$ ,  $t \leq i < h$ , where  $G_h = \text{Gmap}(A_h)$ , and  $G_i$  is obtained from  $G_{i+1}$  by removing arc  $\alpha_{i+1}$  as described in Sec. 3. The two neighbors of  $\alpha_{i+1}$  in  $G_{i+1}$  are recorded as a tentative re-entry point for  $\alpha_{i+1}$  during phase 2. Note that both neighbors may correspond to the same segment, or the segment of one neighbor may coincide with  $s_{\alpha_{i+1}}$ . In phase 2, the algorithm computes incrementally  $G'_i$  and  $\text{FVD}(G'_i)$ , for  $t < i \leq h$ , starting with  $\text{FVD}(G'_t)$ ,  $G'_t = G_t$ .  $G'_t$  is proper as it consists of two maximal arcs.  $G'_{i+1}$  is the arc sequence obtained from  $G'_i$  by inserting back arc  $\alpha_{i+1}$ . During the re-entry of  $\alpha_{i+1}$  a *new* arc may be created. As a result,  $G'_i \neq G_i$ , but  $G'_i$  is proper (by induction and Lemma 2). Because of new arcs, the two recorded neighbors of  $\alpha_{i+1}$  from phase 1 need not be neighbors of  $\alpha_{i+1}$  in  $G'_{i+1}$ ; thus, a number of new arcs may need to be scanned to identify an entry point for  $\alpha_{i+1}$ . The entry point is either an unbounded bisector (regular or artificial), which is deleted from  $\text{FVD}(G'_{i+1})$ , or an arc  $\sigma$ , which entirely contains  $\alpha_{i+1}$ . In the latter case, a new arc is created as  $\text{freg}(\beta)$  splits  $\text{freg}(\sigma)$  in two. At the end of phase 2, we obtain  $\text{FVD}(G'_h) = \text{FVD}(S)$  ( $G'_h = G_h$ ). Details of the two phases are given below.

*Phase 1 (deletion phase).* The deletion of an arc  $\beta = \alpha_{i+1}$  has been described in Sec. 3. Along with the deleted arc  $\beta$ , we record pair  $\alpha, \gamma$ , neighboring  $\beta$  in  $G_{i+1}$ , as a tentative re-entry point for  $\beta$  in phase 2, and a label indicating the position of  $\nu(\alpha, \gamma)$  with respect to  $\beta$ : “within”, “clockwise” or “counterclockwise”. If  $s_\alpha = s_\gamma$ , we keep  $\alpha$  and  $\gamma$  separately, letting  $\nu(\alpha, \gamma) = \nu(s_\beta)$ ; however,  $\alpha\gamma$  is a single maximal arc of  $s_\alpha$ . Note that  $\beta$  must be a segment arc (i.e., it contains  $\nu(s_\beta)$ ), thus,  $\nu(s_\beta)$  must be contained in the expanded  $\alpha\gamma$ . If  $s_\alpha = s_\beta$ , the deleted arc  $\beta$  stores also  $\nu(\alpha, \beta)$  so that it knows its separator from  $\alpha$  when it is inserted back in phase 2. Recall that in this case  $\nu(\alpha, \beta)$  must be the hull direction of an already deleted segment arc. Respectively for  $s_\beta = s_\gamma$ .

*Phase 2 (insertion phase).* First, we compute  $\text{FVD}(G_t)$ . Let  $\alpha, \beta$  be the two maximal arcs constituting  $G_t$ . Each may consist of several consecutive arcs of the same segment. To compute  $\text{FVD}(G_t)$ , we first compute  $\text{FVD}(\alpha\beta)$  in time  $O(1)$ , as illustrated in Figs. 3, 7(a); then we compute  $\text{FVD}(G_t)$  in  $O(t)$  time,

by simply inserting all artificial bisectors between the consecutive sub-arcs of  $\alpha$  (resp.,  $\beta$ ) and splitting  $freg(\alpha)$  (resp.,  $freg(\beta)$ ) accordingly.

Then, we iteratively compute  $FVD(G'_{i+1})$  for  $t < i \leq h$ . Let  $\beta = \alpha_{i+1}$ , and let  $\alpha, \gamma$  be the pair stored with  $\beta$  at phase 1. We provide a case analysis. Let us first assume that  $s_\alpha, s_\gamma \neq s_\beta$ . If  $\alpha$  and  $\gamma$  are still neighbors in  $G'_i$ , then the unbounded Voronoi edge of  $b(\alpha, \gamma)$  (normal or artificial) in  $FVD(G'_i)$  provides an entry point for  $freg(\beta)$ .

Let us assume that  $\nu(\alpha, \gamma)$  is within  $\beta$  (see e.g., Fig. 6(a)). Then  $freg(\beta)$  can be traced over  $b(\alpha, \gamma)$  in time proportional to the complexity of  $freg(\beta)$ , (see, e.g., [9]). Fig. 4 illustrates the insertion of an arc of segment  $s_3$  between two arcs of segment  $s_5$ ; the entry point is the artificial bisector (labeled  $r$ ) separating the regions of the two arcs. Fig. 8 illustrates the insertion of arc  $\epsilon$  between arcs  $\alpha$  and  $\delta$ , where  $s_\alpha \neq s_\delta$  and  $\nu(\alpha, \delta)$  is within  $\epsilon$ ; the entry point is the unbounded bisector  $b(\alpha, \delta)$ , shown shaded in Fig. 8(b).

It is also possible that  $\nu(\alpha, \gamma)$  is not within  $\beta$  (see Fig. 6(b)). This means that in phase 1 arc  $\alpha$  (resp.,  $\gamma$ ) had shrunk during the removal of  $\beta$ , and  $\nu(\alpha, \gamma)$  has been labeled “counterclockwise” (resp., “clockwise”) with respect to  $\beta$ . We consider the case of shrunk  $\alpha$  only as  $\gamma$  is symmetric. In this case,  $freg(\alpha)$  is split in two parts by the insertion of  $freg(\beta)$ , and a new arc  $\alpha'$  for segment  $s_\alpha$  is produced; one part remains  $freg(\alpha)$  and the other becomes  $freg(\alpha')$ . This can be done in time proportional to the complexity of  $freg(\beta)$  plus the complexity of  $freg(\alpha')$ .

If  $\alpha$  and  $\gamma$  are no longer neighbors, a number of new arcs may need to be traced to identify an entry point for  $\beta$ ; this is either a pair of consecutive new arcs  $\delta', \epsilon'$  such that  $\nu(\delta', \epsilon') \in \beta$ , or a single new arc  $\sigma$  that entirely contains  $\beta$ . In the latter case,  $freg(\beta)$  splits  $freg(\sigma)$  into  $freg(\sigma_1)$  and  $freg(\sigma_2)$  in  $FVD(G'_{i+1})$ , which can be done in time proportional to the complexity of  $freg(\beta)$  and  $freg(\sigma_1)$ . In the former case,  $freg(\beta)$  is inserted in time proportional to the complexity of its boundary plus any regions of new arcs that are deleted by being enclosed in  $freg(\beta)$ . Note that a number of consecutive new arcs and their regions may be deleted by the insertion of  $\beta$ , however, this does not affect the overall time complexity, as the number of new arcs is bounded (see Lemma 3). Clearly no original arc can be deleted by the insertion of  $\beta$ .

If  $s_\alpha = s_\beta$  (resp.,  $s_\gamma = s_\beta$ ), then we insert  $\beta$ , by simply splitting arc  $\alpha$  in  $G'_i$  into two parts separated by  $\nu(\alpha, \beta)$ , which is the direction that had been stored with  $\beta$  in phase 1; one part remains as  $\alpha$  and the other becomes  $\beta$ . Recall that  $\nu(\alpha, \beta)$  must be the hull direction of some segment arc  $\alpha_k, k > i + 1$ , which will be inserted later. In addition,  $freg(\alpha)$  in  $FVD(G'_i)$  is split into  $freg(\alpha)$  and  $freg(\beta)$  in  $FVD(G'_{i+1})$ , by inserting the artificial bisector implied by  $\nu(\alpha, \beta)$  (see Def. 2). This can be performed in time proportional to the complexity of  $freg(\beta)$ . Region  $freg(\gamma)$  remains intact.

**Lemma 3.** *The number of arcs in  $G'_i$  is at most  $2i$  and the complexity of  $FVD(G'_i)$  is  $O(i)$ .*

*Proof.* At each step of phase 2, we are creating at most one new arc by the insertion of one original arc. Thus, the total number of arcs in  $G'_i$  ( $|G'_i|$ ) is at

most  $2i$ . Since  $G_t$  consists of exactly two maximal arcs,  $\text{FVD}(G_t)$  must be proper as illustrated in Fig. 3 (for disjoint segments) and Fig. 7(b) (for intersecting segments). Then, using induction and lemma 2,  $\text{FVD}(G'_i)$  must also be proper. But then by Lemma 1,  $\mathcal{T}(G'_i)$  is a tree, with vertices of degree at least three. Since  $\text{FVD}(G'_i)$  has exactly one face for every element of  $G'_i$  and  $|G'_i| \leq 2i$ , the claim follows.  $\square$

**Lemma 4.** *The expected number of new arcs traced at any step of phase 2 is  $O(1)$ .*

*Proof.* By construction,  $G'_i$  is an augmented version of  $G_i$  consisting of at most  $i$  new arcs plus the  $i$  original arcs of  $G_i$ . To insert arc  $\alpha_{i+1} = \beta$  at one step of phase 2, the pair  $\alpha, \gamma$  of consecutive original arcs (that has been stored with  $\alpha_{i+1}$ ) is picked, and some new arcs between  $\alpha$  and  $\gamma$  in  $G'_i$  may be traced. Since every element of  $A_{i+1}$  is equally likely to be  $\alpha_{i+1}$ , each pair of consecutive original arcs in  $G'_i$  has probability  $1/i$  to be considered at step  $i$ . Let  $n_j$  be the number of new arcs in-between the  $j$ th pair of original arcs in  $G'_i$ ,  $1 \leq j \leq i$ ;  $\sum_{j=1}^i n_j \leq i$ . The expected number of new arcs that are traced is then  $\sum_{j=1}^i (n_j/i) \leq 1$ .  $\square$

**Theorem 1.** *Once the farthest hull of a set  $S$  of  $n$  line segments is available, their farthest Voronoi diagram can be computed in additional  $O(h)$  expected time, where  $h$  is the number of faces in  $\text{FVD}(S)$ .*

*Proof.* We use backwards analysis, going from  $\text{FVD}(G'_{i+1})$  to  $\text{FVD}(G'_i)$ . Recall that inserting  $\text{freq}(\alpha_{i+1})$  in  $G'_i$  takes time proportional to the complexity of its boundary, plus occasionally, the complexity of its neighboring Voronoi cell. The latter addition represents a difference from the corresponding argument in the case of points. In addition, it may require the tracing of a number of new arcs between the neighbors of  $\alpha_{i+1}$  that were identified in phase 1. By Lemma 4, the latter time is expected constant. Since  $A_h$  is a random permutation of the arcs in  $G_h$ , the expected time complexity of inserting  $\text{freq}(\alpha_{i+1})$  in  $\text{FVD}(G'_i)$  is equivalent to the expected complexity of a randomly selected face in  $\text{FVD}(G'_{i+1})$ , plus the expected complexity of its immediate neighbor. Since  $\text{FVD}(G'_i)$  has size  $O(i)$  and it consists of  $O(i)$  faces, the expected complexity of a randomly selected region is constant. The same holds for one neighbor of the randomly selected region. Thus, the expected time spent to insert  $\text{freq}(\alpha_{i+1})$  in  $\text{FVD}(G'_i)$  is constant. Since the total number of arcs is  $h$ , the claim follows.  $\square$

## 5 A Deterministic Linear Divide-and-Conquer Algorithm

We now augment the framework of Aggarwal et al. [1] for points in convex position with techniques from Secs. 3, 4, and derive a linear-time algorithm to compute  $\text{FVD}(S)$ , given  $\text{Gmap}(S)$ . Let  $G$  be a subsequence of  $\text{Gmap}(S)$ , and let  $G'$  be a corresponding proper augmented subsequence such that the complexity of  $G'$  is  $O(|G|)$ , where  $|G|$  denotes the number of arcs of the sequence  $G$ . Our algorithm follows the flow of [10], which in turn follows [1].

1. Simplify  $G$  by uniting any consecutive arcs of the same segment into a single maximal arc for that segment.
2. Color each arc of  $G$  *red* or *blue* by applying the following two rules:
  - (a) For each 5-tuple  $F$  of consecutive arcs  $\alpha\beta\gamma\delta\epsilon$  in  $G$ , compute  $\text{FVD}(F')$  as follows: start with  $\text{FVD}(F_2)$ , where  $F_2 = \gamma\delta$ , and consecutively insert the arcs  $\beta, \epsilon, \alpha$  in this order (see Sec. 3). Color  $\gamma$  red, if  $\text{freq}(\gamma)$  in  $\text{FVD}(F')$  does not neighbor the region of any arc of segment  $s_\alpha$  nor of segment  $s_\epsilon$ . Otherwise, color  $\gamma$  blue.
  - (b) For each series of consecutive blue arcs, color red every other arc except the last arc.
3. Let  $B$  (blue) be the sequence obtained from  $G$  by deleting all the red arcs. Recursively compute  $\text{FVD}(B')$ . ( $B'$  is a possibly augmented version of  $B$ .)
4. Partition the red arcs into *crimson* and *garnet* as follows. Re-color as *crimson* at least a constant fraction of the red arcs, such that for any two crimson arcs, if they were inserted in  $\text{FVD}(B')$ , the closures of their Voronoi regions would be disjoint.
5. Insert the crimson arcs one by one in  $\text{FVD}(B')$  resulting in  $\text{FVD}(V')$ .
6. Let  $Gr$  (garnet) be the sequence obtained from  $G$  by deleting all blue and crimson arcs. Recursively compute  $\text{FVD}(Gr')$ .
7. Merge  $\text{FVD}(V')$  and  $\text{FVD}(Gr')$  into  $\text{FVD}(G')$  so that  $|G'|$  is  $O(|G|)$ .
8. For any arcs that were united in Step 1, subdivide their regions in  $\text{FVD}(G')$  into finer parts by inserting the corresponding artificial bisectors.

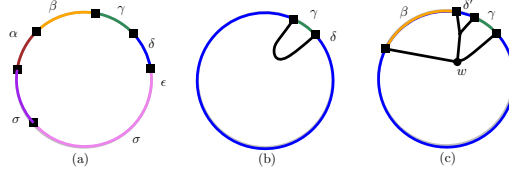
The recursion ends when the number of maximal arcs in  $G$  is at most five. In this case  $\text{FVD}(G')$  is directly computed in  $O(1)$  time and then enhanced as indicated in Step 8. If all arcs in  $G$  are of the same segment, no diagram is computed but instead  $G$  is returned as a simple list of arcs. If the recursive call at Step 6 returns such a list, then in Step 7 we obtain  $\text{FVD}(G')$  by inserting these arcs in  $\text{FVD}(V')$  one by one. Note, that the recursive call at Step 3 always returns  $\text{FVD}(B')$ . In the following we describe the various steps in more detail.

*Step 2.* Rules 2a and 2b guarantee that no two consecutive arcs in  $G$  are red and no three consecutive arcs in  $G$  are blue. This statement can be proven following the spirit of [10, Lemma 8]; however, the argument is more involved, because of the appearance of new arcs (see Lemma 5).

**Lemma 5.** *No two consecutive arcs in  $G$  are red and no three consecutive arcs in  $G$  are blue.*

*Proof.* Suppose for the sake of contradiction, that there are two consecutive arcs  $\gamma, \delta$  in  $G$ , which are both colored red. Clearly, they are both colored red by Rule 2a (not by Rule 2b). Thus, there is a 6-tuple  $\alpha\beta\gamma\delta\epsilon\sigma$  in  $G$  such that  $\gamma$  and  $\delta$  satisfy Rule 2a. Since all arcs in  $G$  are maximal, any two consecutive ones are of different segment. Then  $s_\beta \neq s_\delta$  and  $s_\gamma \neq s_\epsilon$ , as otherwise the respective regions would be not neighboring.

Consider the 5-tuple  $F = \alpha\beta\gamma\delta\epsilon$  (see Fig. 9(a)), and  $\text{FVD}(F')$ , derived by applying Rule 2a. Building  $\text{FVD}(F')$  starts from  $\text{FVD}(F'_2)$ , shown schematically in



**Fig. 9.** (a) a 5-tuple  $F$ ; scheme of (b)  $\text{FVD}(F'_2)$ , (c)  $\text{FVD}(F'_3)$

Fig. 9(b). The insertion of  $\beta$  results in sequence  $F'_3$ , which is either  $\beta\gamma\delta$ , or  $\beta\delta'\gamma\delta$  (see Fig. 9(c)). Since  $\gamma$  is colored red by Rule 2a,  $\text{freq}(\gamma)$  borders only regions of arcs of segments  $s_\delta$  and  $s_\beta$ . Since the arc bisector  $b(\gamma, \delta)$  is an unbounded portion of  $b(s_\gamma, s_\delta)$ , the (oriented) bisector  $b(s_\gamma, s_\delta)$  must meet  $b(s_\beta, s_\gamma)$  first (before it meets  $b(s_\delta, s_\epsilon)$ ) forming a Voronoi vertex (vertex  $w$  on Fig. 9(c)). Bisectors are oriented from the point at infinity incident to their relevant arc bisectors. By a similar argument for the 5-tuple  $D = \beta\gamma\delta\epsilon\sigma$ , the oriented  $b(s_\delta, s_\epsilon)$  must meet  $b(s_\gamma, s_\delta)$  before any other involved bisector, i.e.,  $b(s_\gamma, s_\delta)$  meets  $b(s_\delta, s_\epsilon)$  before  $b(s_\beta, s_\gamma)$ . A contradiction.

Rule 2b prevents three consecutive blue arcs.  $\square$

*Steps 4 and 5.* To chose the crimson arcs we apply the *combinatorial lemma* of [1]. The lemma states that for a binary tree  $T$  with  $n$  leaves embedded in  $\mathbb{R}^2$ , if each leaf of  $T$  is associated with a subtree of  $T$  and for any two successive leaves these subtrees are disjoint, then in  $O(n)$  time we can chose a set of leaves, whose number is at least a constant fraction of  $n$  and whose subtrees are pairwise disjoint. We associate each red arc  $\beta$  with a unique leaf of  $\mathcal{T}(B')$ , which would be an entry point for  $\beta$  if  $\beta$  were inserted in  $\text{FVD}(B')$ . If the insertion of  $\beta$  in  $B'$  splits an arc of  $B'$  in two, then we add an artificial bisector to  $\mathcal{T}(B')$  to serve as a potential entry point for  $\beta$  in Step 5. The modified  $\mathcal{T}(B')$  satisfies the requirements of the combinatorial lemma (see Lemma 6, similarly to [10, Lemma 9]) and has complexity  $O(|B'| + |R|)$ , where  $|R|$  denotes the number of red arcs, as we insert at most one artificial bisector per red arc. The time spent in Step 4 is also  $O(|B'| + |R|)$ . The added artificial bisectors serve as entry points in  $\text{FVD}(B')$  for the crimson arcs in Step 5. The complexity of the resulting sequence  $V'$  in Step 5 is set in the proof of Lemma 9.

The following lemma ensures that the combinatorial lemma of [1] can be applied to  $\mathcal{T}(B')$ , similarly to [10, Lemma 9]).

**Lemma 6.** *For any two successive red arcs in  $G$ , if they were inserted in  $\text{FVD}(B')$ , the closures of their Voronoi regions would be disjoint.*

*Proof.* Suppose for the sake of contradiction that there is a pair of successive red arcs in  $G$  which do not satisfy the statement. By Rule 2b, there is either one or two blue arcs between them in  $G$ . In the former case, there is a 5-tuple  $F = \alpha, \beta, \gamma, \delta, \epsilon$  of the arcs in  $G$ , such that the arcs  $\beta$  and  $\delta$  are colored red, and  $\text{freq}(\beta)$  and  $\text{freq}(\delta)$  in are adjacent if  $\beta$  and  $\delta$  are inserted in  $\text{FVD}(B')$  (by the insertion process of Sec. 3). Then the regions of  $\beta$  and  $\delta$  must be adjacent in  $\text{FVD}(F')$ , and they are obtained from  $F$  as described in Rule 2a. Then  $\gamma$  must

satisfy Rule 2a and it must have been colored red during Step 2; a contradiction. If there are two blue arcs between the red arcs in question, the same argument works for one of the two corresponding 5-tuples.  $\square$

*Step 7.* In Step 7 we obtain  $G'$  and  $\text{FVD}(G')$  by merging  $\text{FVD}(A)$  and  $\text{FVD}(B)$ , where  $A = V'$ ,  $B = Gr'$ . We cannot perform an ordinary merging because the complexity of  $\text{FVD}(G')$  may grow too much due to the appearance of new arcs. To keep the complexity of  $G'$  within  $O(|G|)$ , we merge the two diagrams while discarding parts guaranteed to contain no original arcs.

Merging two Voronoi diagrams requires typically two standard steps: (1) identify starting points for the *merge curves*; and (2) trace the *merge curves*. A merge curve is a sequence of Voronoi edges in the resulting diagram whose incident regions come from different original diagrams. Here, we identify starting points (part (1)) only for merge curves related to original arcs. Skipping a merge curve has the effect of discarding the portion of one diagram, which is bounded by it. Tracing (part (2)) can then be performed in the standard way (see, e.g., [9]).

To illustrate part (1) we use the dual space for simplicity. Recall that an arc sequence corresponds to a pair of  $x$ -monotone paths in the dual arrangement of lower and upper wedges. Let  $U$  (resp.,  $L$ ) be the upper (resp., lower) envelope of  $A$  and  $B$  in the arrangement of lower (resp., upper) wedges.  $U$  (resp.,  $L$ ) consists of a series of connected components alternating between the paths of  $A$  and  $B$ . The vertices incident to such connected components reveal the starting points of merge curves, which can be easily identified in time  $O(|A| + |B|)$ . Each merge curve has exactly two starting points in  $U \cup L$ . We identify all such starting points, and among them we chose only those whose incident component of  $B$  contains original arcs. For each merge curve of interest either one or two starting points are identified. If for a merge curve no starting points are identified then

it can be safely ignored, because the incident component of  $B$  contains no original arcs and it will not appear in  $\text{FVD}(S)$ . The description can be adapted to work directly on the arc sequences  $A, B$  similarly to [13].

The following two lemmas establish the correctness of the merging process. Recall that  $G'$  is the arc sequence of  $\text{FVD}(G')$  which is obtained by merging  $\text{FVD}(V')$  and  $\text{FVD}(Gr')$  using the above two step process.

**Lemma 7.**  $|G'|$  is  $O(|A| + |B|)$ . Moreover,  $G'$  contains all the original arcs of  $A$  and  $B$ , and only  $O(|G|)$  new arcs, which are neither in  $A$  nor in  $B$ .

*Proof.* (a) Consider the arrangement of lower wedges and its upper envelope  $U$  (the case of upper wedges is symmetric). Clearly, every original arc in  $A$  and  $B$  must entirely contain its corresponding arc in  $U$ .  $G'$  in dual space is a path, which lies above or on (but never below) the path dual to  $A$ , thus all original arcs from  $A$  are in  $G'$ . An original arc  $\beta \in B$  must be part of a maximal component of  $B$  in  $U$ . For each such component, the algorithm for (1) chooses the starting points of both merge curves incident to this component. Thus,  $\beta \in G'$ .

(b) An arc that is neither in  $A$  nor in  $B$  is created when a portion of  $B$  in  $G'$  splits an arc of  $A$  (or vice versa). Thus, the number of such new arcs in  $G'$  is



bounded from above by the number of maximal components of  $B$  and of  $A$  that are contained in  $G'$ . This is in turn bounded by the number of original arcs in  $A$  and  $B$ , which is  $|G|$ .  $\square$

By Lemma 7, identifying the starting points of the merge curves is performed in time  $O(|G'|)$ . The merge curves are traced in a standard way also in  $O(|G'|)$  time. Thus, the entire merging process (Step 7) is performed in time  $O(|G'|)$ .

**Lemma 8.** *Suppose that  $V'$  and  $Gr'$  are proper. Then  $\text{FVD}(G')$ , obtained by merging  $\text{FVD}(V')$  and  $\text{FVD}(Gr')$ , is also proper.*

*Proof.* It is enough to show that any merge curve consists solely of arc bisectors. Tracing the merge curve inside a single region  $\text{freg}(\alpha)$  of  $\text{FVD}(V')$  (resp.,  $\text{FVD}(Gr')$ ) is equivalent to tracing the boundary of  $\text{freg}(\alpha)$  if it would be inserted in  $\text{FVD}(Gr')$  (resp.,  $\text{FVD}(V')$ ). Thus, we can repeatedly apply the argument of the proof of Lemma 2 to any of the merge curves.  $\square$

*Time complexity.* Step 4 is performed by applying the combinatorial lemma of [1]. Because of Step 2 (Lemma 5), there exists a positive constant  $q_1$  such that  $|B| \leq q_1|G|$ , and by the combinatorial lemma there exists a positive constant  $q_2$  such that  $|Gr| \leq q_2|G|$ , where  $q_1 + q_2 < 1$ . As a result,  $|G'|$  remains  $O(|G|)$  as shown in the following lemma.

**Lemma 9.**  *$|G'|$  is  $O(|G|)$ .*

*Proof.* Let  $S(m)$  denote the maximum number of arcs in  $G'$  as a function of  $m = |G|$ . We show that  $S(m) \leq S(q_1m) + S(q_2m) + O(m)$ , and thus  $S(m) = O(m)$ . Indeed,  $\text{FVD}(G')$  is obtained by insertion of the crimson arcs into  $\text{FVD}(B')$ , and by merging the result with  $\text{FVD}(Gr')$ . By Lemma 7,  $|G'| \leq |V'| + |Gr'| + O(m)$ . The number of crimson arcs is  $O(m)$ , and the insertion of each of them causes at most one new arc, thus, the number of arcs in  $V'$  is at most  $S(q_1m) + O(m)$ .  $\square$

**Theorem 2.** *Given  $\text{Gmap}(S)$ , the  $\text{FVD}(S)$  can be computed in additional  $O(h)$  time, where  $h$  is the number of faces in  $\text{FVD}(S)$ .*

*Proof.* By Lemma 9, the time complexity of the algorithm can be analyzed similarly to [1]. Let  $T(m)$  denote the time to compute  $\text{FVD}(G')$ , where  $|G| = m$ . By Lemma 9, the complexity of the diagrams remains linear in the number of their original arcs. Thus, Step 7 is performed in time  $O(m)$ . Choosing the crimson arcs and inserting them in  $\text{FVD}(B')$  is also done in time  $O(m)$  (see Step 4). Recall that all entry points for the crimson arcs are determined in Step 4 by modifying  $\mathcal{T}(B')$  in time  $O(m)$ . Therefore, the algorithm satisfies the recursive inequality of [1],  $T(m) \leq T(q_1m) + T(q_2m) + O(m)$ , which implies  $T(m) = O(m)$ .  $\square$

## 6 Computing the Order- $(k+1)$ Subdivision within an Order- $k$ Voronoi Region

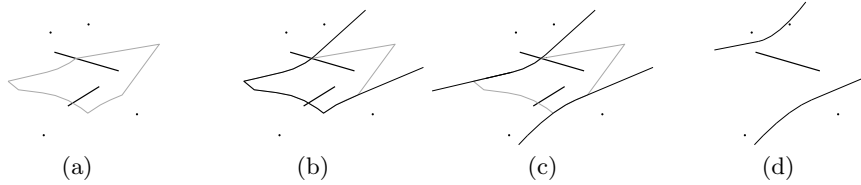
Let  $F$  be a face of the order- $k$  Voronoi region  $k\text{-reg}(H)$ ,  $H \subset S$ ,  $|H| = k$ . Let  $S_F \subseteq S \setminus H$  consist of segments that induce the boundary  $\partial F$ . Consider the order-1 Voronoi diagram of  $S_F$ , within  $F$ ,  $\mathcal{V}_1(S_F)$ . As shown recently [14],  $\mathcal{V}_1(S_F)$  is

a tree structure, and the sequence of its faces along  $\partial F$  forms a Davenport-Schinzel sequence of order 4 (order 2 for non-intersecting segments). We can compute  $\mathcal{V}_1(S_F)$  in time linear in the complexity of  $\partial F$  by slightly adapting the algorithms in the previous sections. This directly implies that the order- $k$  Voronoi diagram of  $S$  can be computed in  $O(k^2n + n \log n)$  time, improved from  $O(k^2n \log n)$ , by iteratively computing higher-order diagrams, starting at  $\mathcal{V}_1(S)$ .

Consider the sequence  $A$  of segment appearances along  $\partial F$ . Each appearance of  $s \in S_F$  corresponds to an arc  $\alpha$  of  $\partial F$ , as delimited by two consecutive Voronoi vertices. Arc  $\alpha$  is *visible* from  $s$ , in the sense that for any point  $x$  on  $\alpha$ , the segment realizations  $d(s, x)$  does not intersect  $\partial F$ . The sequence  $A$  is represented by the arcs of  $\partial F$  (and vice versa).

Let  $A_i, i \leq h$  denote an arbitrary collection of  $i$  elements from  $A$ . Let  $F_h = F$ .  $F_i$  can be regarded as the Voronoi region of a cluster  $H$ ,  $|H| = k$ , in a *Hausdorff Voronoi diagram* (also called *cluster Voronoi diagram* [3]) of the family of clusters  $C_i = \{H, \{\alpha_i\}, \alpha_i \in A_i\}$ .

The deletion process computes  $\partial F_i$  from  $\partial F_{i+1}$  by deleting  $\alpha_{i+1}$  and recomputing the Hausdorff region of  $H$ . In particular, let  $\alpha, \beta, \gamma$  be arcs in counter-clockwise order along  $\partial F_{i+1}$ . To delete  $\beta$  we remove its incident Voronoi vertices and extend the incident Voronoi edges on  $\partial F_{i+1}$  inducing  $\alpha$  and  $\gamma$ , until they intersect, creating a new Voronoi vertex, or until it is determined that the incident Voronoi edges become unbounded.  $F_i$  entirely encloses  $F_{i+1}$  and it may become unbounded. Once unbounded,  $\partial F_i$  may break into two connected components by the deletion of one additional arc (see Lemma 10). Figure 10(a)-(c) shows how  $\partial F_i$  changes as a result of removing of elements from the boundary of an order-2 Voronoi region.



**Fig. 10.** Two line segments and (a) their 2-order Voronoi region; (b) the first opening of the boundary after removing two elements (c) the disconnection of the boundary into two connected components; (d) removing a segment does not reduce the number of connected components of the Hausdorff Voronoi region

The insertion process inserts back the elements of  $A_i$  in reverse order and computes (the augmented)  $F'_i, A'_i$ , and the corresponding Voronoi diagram  $\mathcal{V}_1(A'_i)$ . New arcs may be created similarly to Secs. 4, 5.

**Lemma 10.**  $\partial F_i$  may consist of at most two connected components.

*Proof.* By induction on  $k$ , the number of line segments in cluster  $H$ . If  $H$  is a singleton ( $k = 1$ ), the claim is easy to see. Suppose the claim is true for a cluster  $H$  of  $k$  segments. Assume for the sake of contradiction that there is a cluster  $H$

of  $k + 1$  line segments, for which there exist a set of singleton-neighbors  $N$  such that the boundary of the Hausdorff region of  $H$  has at least three connected components. Each of the connected components separates  $H$  from at least one element of  $N$ . Now, if we remove a line segment from  $H$ , the Hausdorff region of  $H$  can only grow (see Figure 10(c,d)), thus, no two connected components of its boundary may unite. No connected component may disappear, since the singletons in  $N$ , which are separated from  $H$  by these connected components, remain. Thus, the number of connected components remains at least three for a cluster of size  $k$ , a contradiction.  $\square$

If during the deletion process  $\partial F_i$  breaks in two components then we are initially computing two different overlapping diagrams, one for each component of  $\partial F_i$ . When the arc that caused the splitting is re-inserted, these diagrams are merged into one. We give more details in the following.

Suppose that during the deletion process  $\partial F_i$  breaks in two components. Let  $F_i^1$  (resp.,  $F_i^2$ ) denote the portion of the plane bounded by the first (resp., second) component of  $\partial F_i$ , which contains  $F_i$ . If  $\partial F_i$  consists only of one component, let  $F_i^1 = F_i$  and  $F_i^2 = \emptyset$ . Let  $*$  = 1, 2. Then let  $A_i^*$  denote the elements of  $A_i'$  defining  $\partial F_i^*$ . We can define the Voronoi diagram of  $A_i^*$  in  $F_i^*$ , denoted  $\mathcal{V}_1(A_i^*)$ , similarly to Sec. 3.

In particular, a point  $x$  in  $F_i^*$  is said to be *attainable* from  $\alpha$  in  $A_i^*$ , if  $d(s_\alpha, x)$  is realized as a segment passing through  $\alpha$ . Let  $d(\alpha, x) = d(s_\alpha, x)$ , if  $x$  is attainable from  $\alpha$ , and  $d(\alpha, x) = \infty$ , otherwise.  $\mathcal{V}_1(A_i^*)$  is a partitioning of  $F_i^*$  into regions defined for  $\alpha$  in  $A_i^*$ :  $reg(\alpha) = \{x \in F_i^* \mid d(x, \alpha) < d(x, \gamma), \forall \gamma \in A_i^*, \gamma \neq \alpha\}$ . The graph structure of  $\mathcal{V}_1(A_i^*)$  is a collection of trees. If we consider an artificial point at infinity and connect to it all unbounded bisectors, including the unbounded sides of  $\partial F_i^*$ , then  $\mathcal{V}_1(A_i^*)$  is a tree.

An arc bisector  $b(\alpha, \gamma)$  ( $s_\alpha \neq s_\gamma$ ) is a semicurve derived from  $b(s_\alpha, s_\gamma)$  by breaking it at the point of minimum distance from  $s_\alpha, s_\gamma$ , and choosing the part that leaves  $\alpha$  and  $\gamma$  at opposite sides. For  $s_\alpha = s_\gamma$ , we can define an artificial bisector  $b(\alpha, \gamma)$  using a point  $x$  on an arc  $\beta$  between  $\alpha$  and  $\gamma$  visible from  $s_\alpha$ . Let  $b(\alpha, \gamma)$  be the ray  $r$  emanating from  $s_\beta$ , realizing  $d(s_\beta, x)$ , extending towards infinity away from  $s_\beta$ .

Both the randomized and the deterministic algorithm can now be directly applied obtaining a linear construction algorithm for  $\mathcal{V}_1(F)$ . There are two important arc removals and re-insertions. A simple one that converts  $\partial F_i$  into an open curve; and a crucial one that splits the open  $\partial F_i$ , into two components (if any). The re-insertion of this arc, requires time linear in the size of  $A_i^1$  and  $A_i^2$  to merge the two diagrams. In the randomized algorithm this is performed once. Similarly in the deterministic algorithm, an arc can participate to such an operation at most once (see Theorem 3).

Note that  $\partial F_i^*$  need not be explicitly computed. Any arc bisector incident to  $\partial F_i^*$  intersects it after its origin, and no two neighboring such bisectors can intersect before  $\partial F_i^*$ . Thus, we can create the tree structure of  $\mathcal{V}_1(A_i^*)$  without computing its enclosing boundary.

**Theorem 3.** *The order- $(k + 1)$  subdivision in a face  $F$  of the order- $k$  Voronoi diagram can be computed in time proportional to the complexity of  $\partial F$ .*

*Proof.* We only clarify some technical details that differ from Sec. 5. In Step 4 we apply the combinatorial lemma of [1] to  $\mathcal{T}(A_i^*)$ , which denotes the graph structure of  $\mathcal{V}_1(A_i^*)$ . Recall that the root of  $\mathcal{T}(A_i^*)$  is an artificial point at infinity of arbitrary degree  $d$ , however, the combinatorial lemma assumes that all internal nodes are of degree 3. We can easily enforce this condition by inserting  $d - 1$  artificial nodes of degree 3 each that connect the children of the root. The artificial nodes have no effect in the algorithm that applies the combinatorial lemma (see [10]), and Lemma 6 is satisfied. This operation at most doubles the nodes of the tree, and thus, it has no effect in the complexity of the algorithm.

Another difference is the rebuilding operation, which in Step 7 may unite two connected components of  $\partial F_i$ , and their Voronoi diagrams, into one. In the recursion tree of the deterministic algorithm, each path from the root to a leaf corresponds to a specific permutation  $A_h$  of the set  $A$ . By Lemma 10, for any permutation of  $A$  (and thus for any path in the recursion tree) there is at most one arc that disconnects  $\partial F_i$  into two connected components. Each original arc corresponds to one leaf of the recursion tree, and thus, to only one such path. Thus, each original arc participates in at most one of the expensive rebuilding operations. Each rebuilding operation takes time linear in the number of arcs participating in it, which is in turn linear in the number of original arcs involved. Therefore, in total during the whole execution of the algorithm, the rebuilding operations take time  $O(h)$ .  $\square$

Theorem 3 includes updating a nearest-neighbor segment Voronoi diagram after the deletion of one segment in time proportional to the complexity of the deleted region.

## 7 Concluding Remarks

We presented linear-time algorithms to compute tree-like Voronoi diagrams with disconnected regions, after the sequence of their faces along an enclosing boundary (or at infinity) is known. In this paper we considered line segments, however, the presented techniques are not necessarily specific to them. For example, the constructions in this paper apply also to the respective abstract Voronoi diagrams, on which we will report in a companion paper. Note that the farthest abstract Voronoi diagram can be constructed in expected  $O(n \log n)$  time by a randomized incremental construction [11], which is not related to the randomized linear-time approach in this paper. The approach in this paper of first computing the cyclic sequence of faces at infinity, followed by constructing the diagram, results in an  $O(n \log n)$  deterministic algorithm also for the farthest abstract Voronoi diagram.

## References

1. Aggarwal, A., Guibas, L., Saxe, J., Shor, P.: A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geom.* 4, 591–604 (1989)
2. Aurenhammer, F., Drysdale, R., Krasser, H.: Farthest line segment Voronoi diagrams. *Inform. Process. Lett.* 100, 220–225 (2006)
3. Aurenhammer, F., Klein, R., Lee, D.T.: *Voronoi Diagrams and Delaunay Triangulations*. World Scientific (2013)
4. Bohler, C., Cheilaris, P., Klein, R., Liu, C.H., Papadopoulou, E., Zavershynskyi, M.: On the complexity of higher order abstract Voronoi diagrams. In: *Proc. 40th ICALP. LNCS*, vol. 7965, pp. 208–219 (2013)
5. Bohler, C., Klein, R., Liu, C.: Forest-like abstract Voronoi diagrams in linear time. In: *Proc. 26th CCCG* (2014)
6. Cheong, O., Everett, H., Glisse, M., Gudmundsson, J., Hornus, S., Lazard, S., Lee, M., Na, H.: Farthest-polygon Voronoi diagrams. *Comput. Geom.* 44(4), 234–247 (2011)
7. Chew, L.P.: Building Voronoi diagrams for convex polygons in linear expected time. Tech. rep., Dartmouth College, Hanover, USA (1990)
8. Chin, F., Snoeyink, J., Wang, C.A.: Finding the medial axis of a simple polygon in linear time. *Discrete Comput. Geom.* 21(3), 405–420 (1999)
9. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edn. (2008)
10. Klein, R., Lingas, A.: Hamiltonian abstract voronoi diagrams in linear time. In: *9th ISAAC. LNCS*, vol. 834, pp. 11–19 (1994)
11. Mehlhorn, K., Meiser, S., Rasch, R.: Furthest site abstract Voronoi diagrams. *Int. J. Comput. Geom. Ap.* 11(6), 583–616 (2001)
12. Papadopoulou, E.: Net-aware critical area extraction for opens in VLSI circuits via higher-order Voronoi diagrams. *IEEE T. Comput. Aid. D.* 30(5), 704–716 (2011)
13. Papadopoulou, E., Dey, S.K.: On the farthest line-segment Voronoi diagram. *Int. J. Comput. Geom. Ap.* 23(6), 443–459 (2013)
14. Papadopoulou, E., Zavershynskyi, M.: The higher-order Voronoi diagram of line segments. *Algorithmica* DOI 10.1007/s00453-014-9950-0 (2014)